

09842562,032101

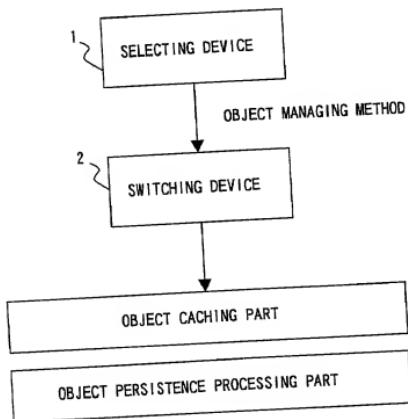


FIG. 1

107260-29527360

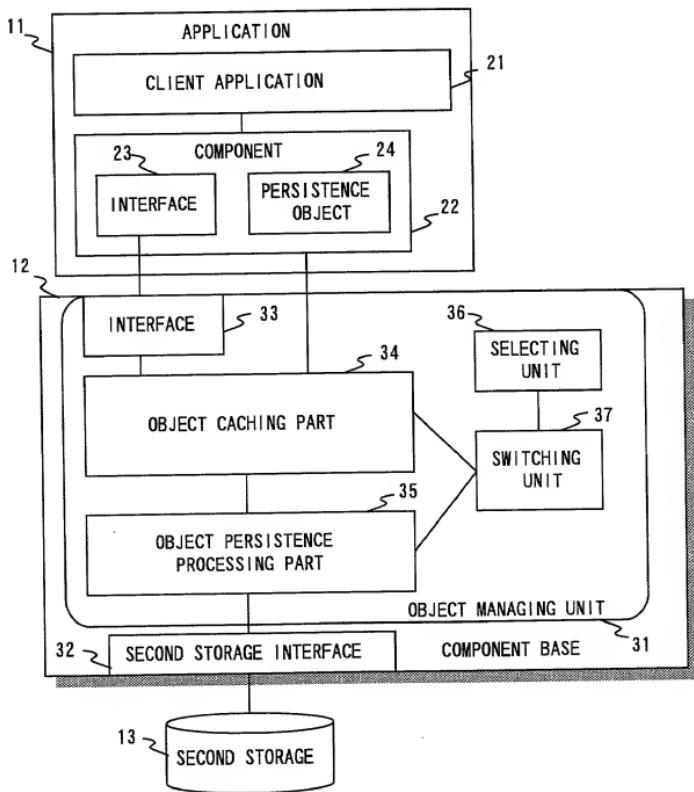


FIG. 2

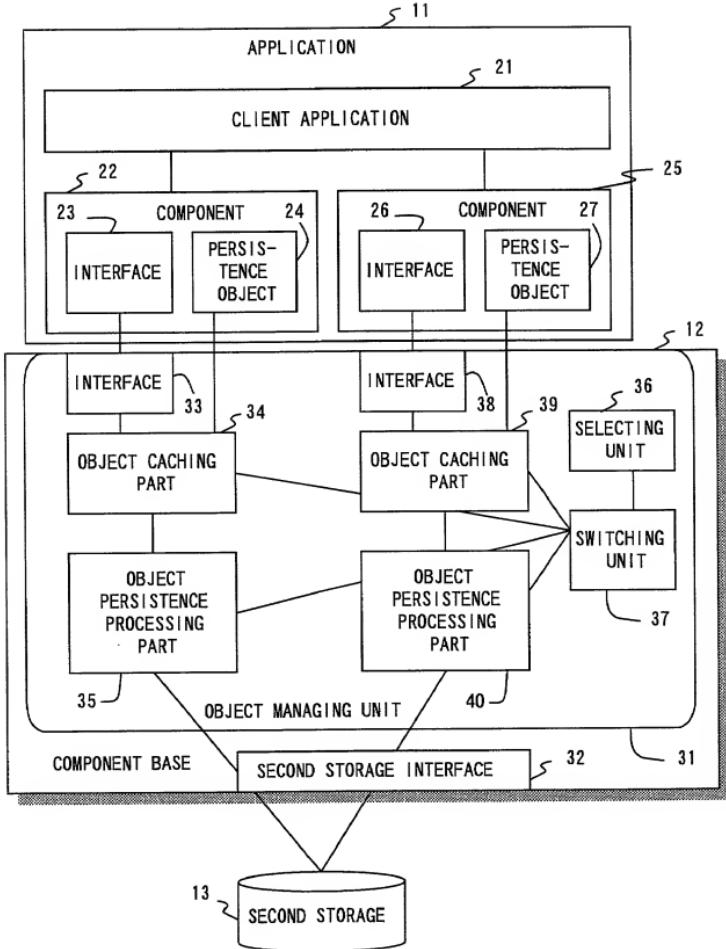
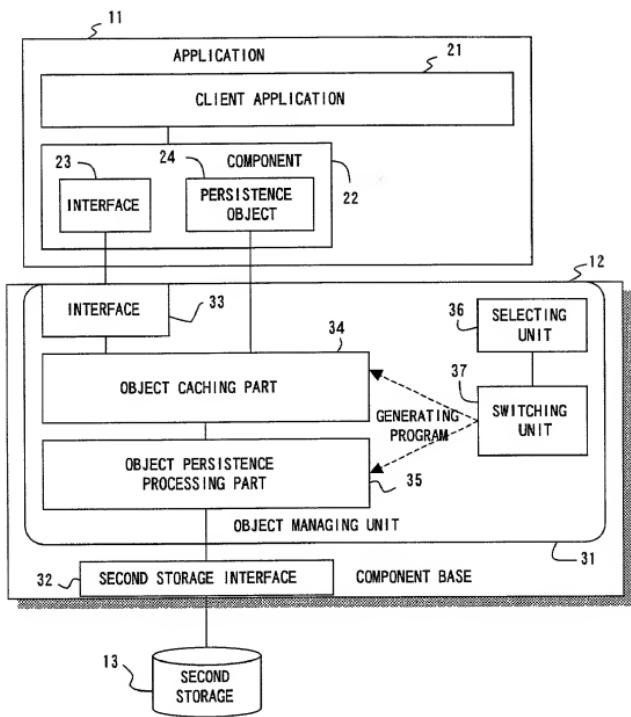


FIG. 3

00012552-032104



F I G. 4

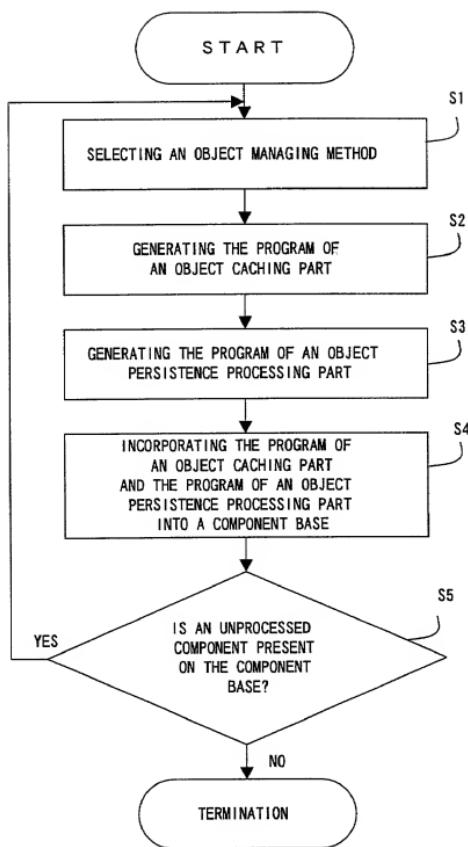


FIG. 5

1012003905360

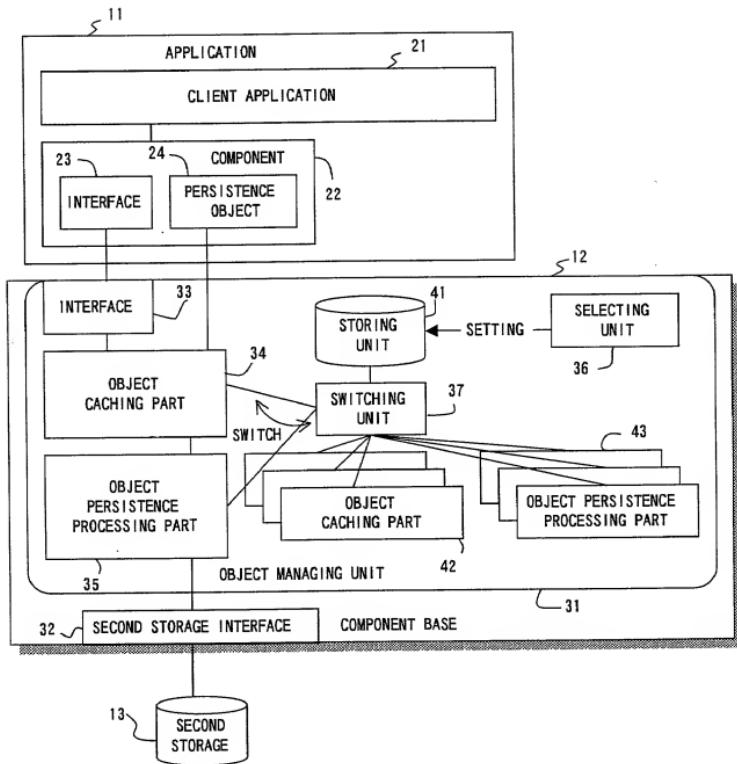


FIG. 6

09842562 032101

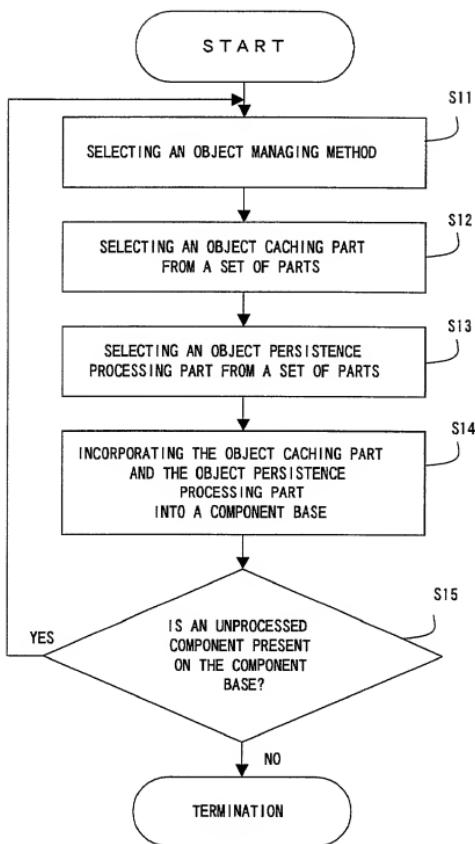


FIG. 7

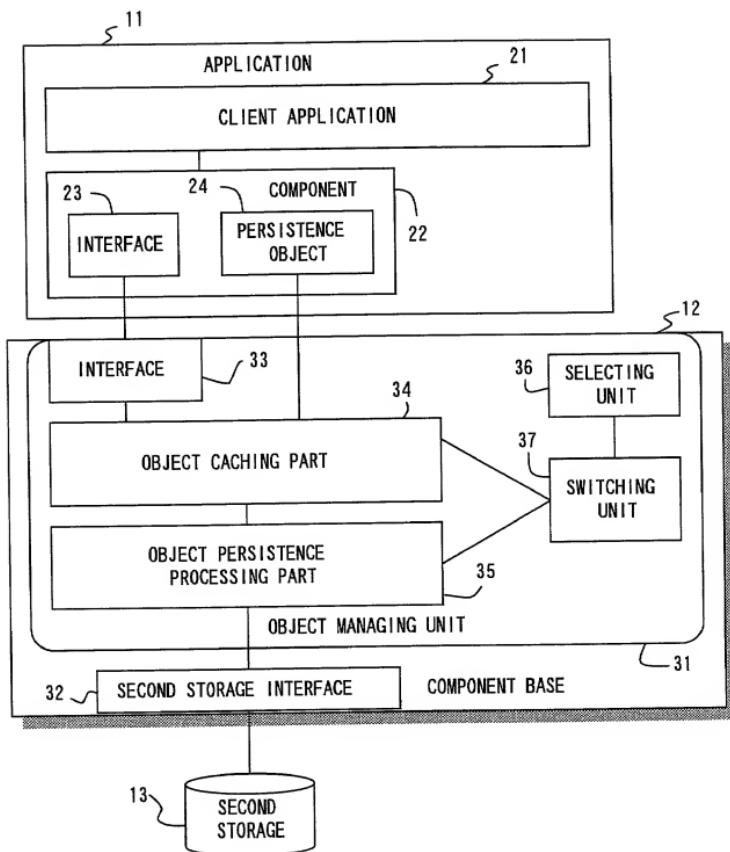
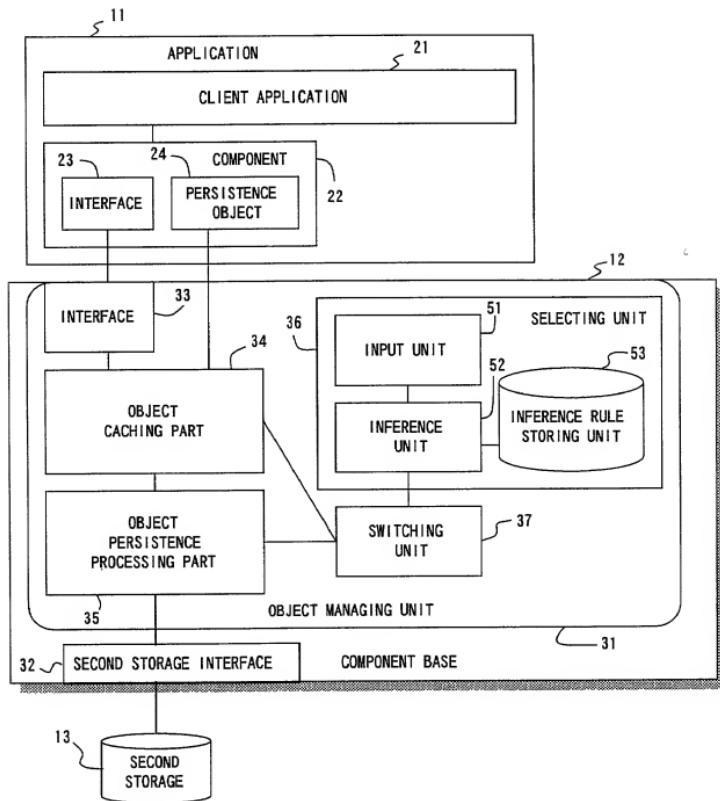
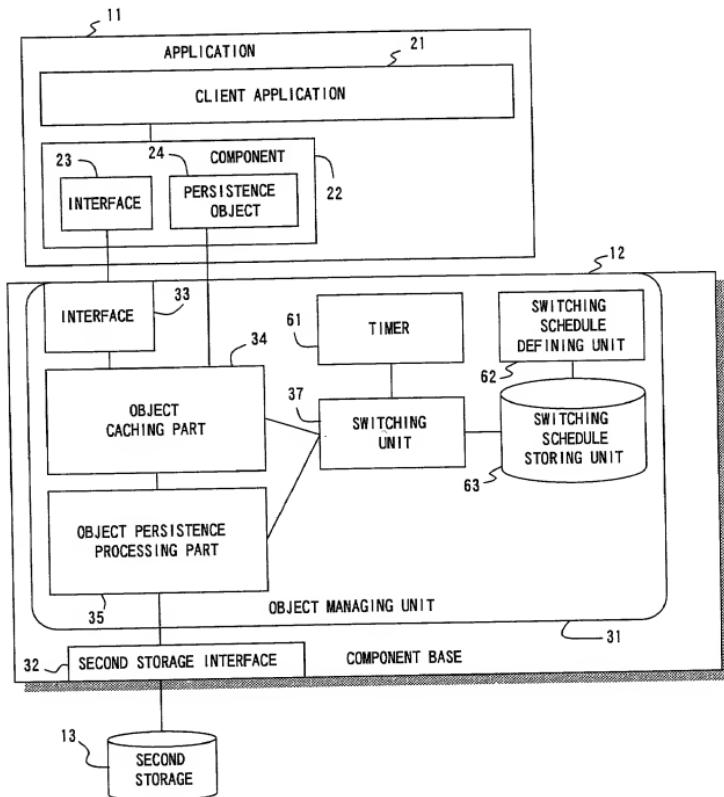


FIG. 8

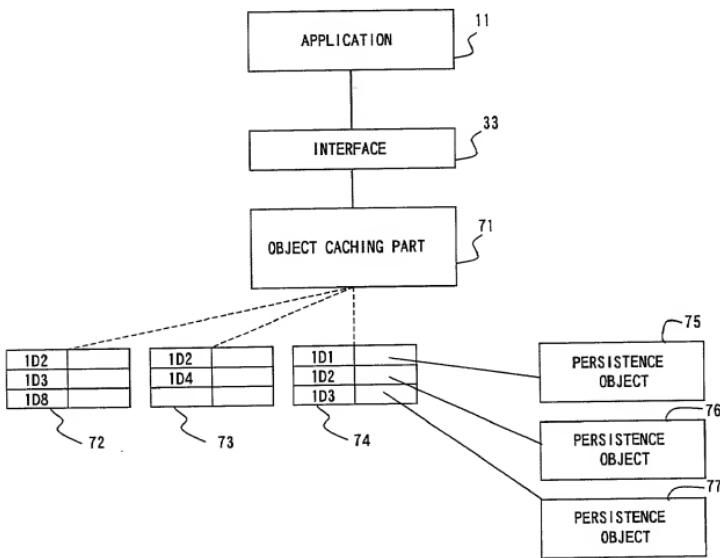


F I G. 9

102501-29227360



F I G. 1 0



F I G. 1 1

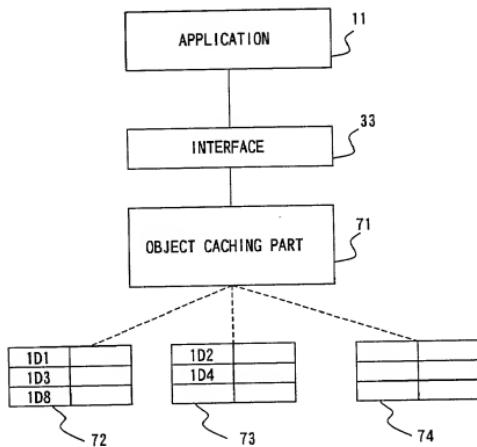
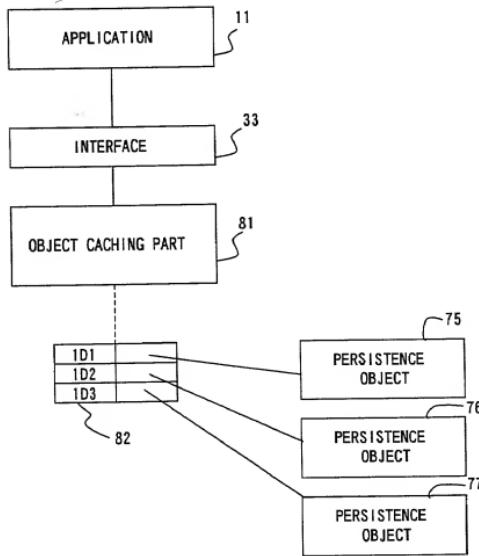


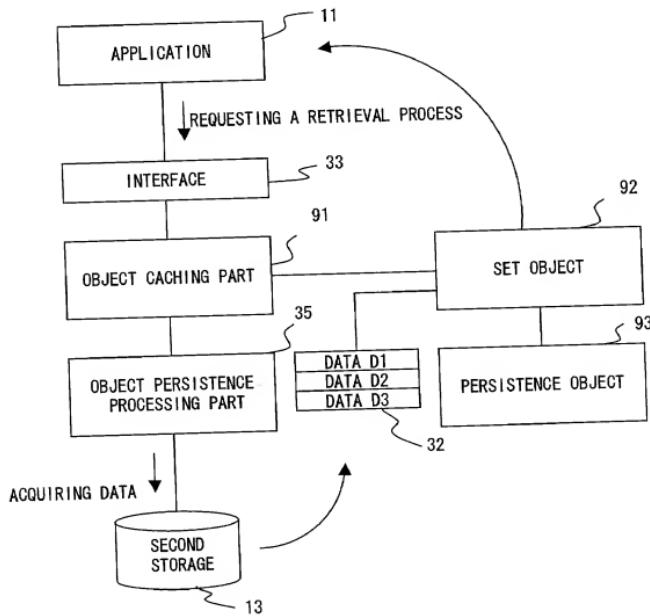
FIG. 12

00812552 032104



F I G. 1 3

00142562-0220104



F I G. 1 4

09342953 - 0932104

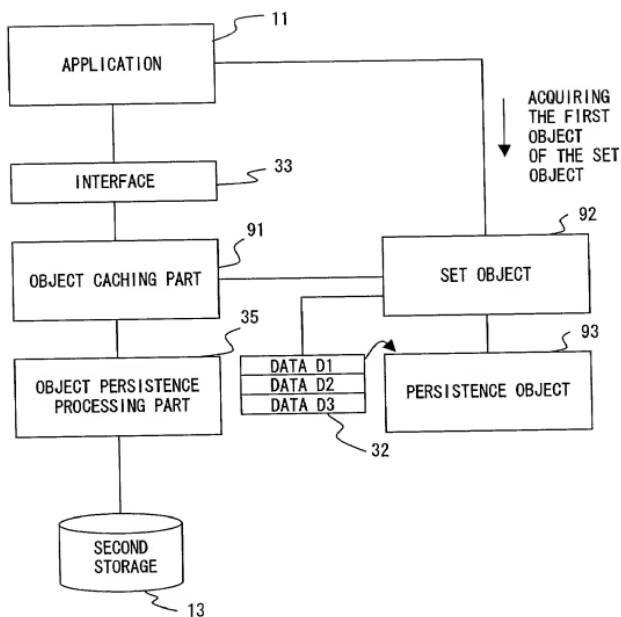


FIG. 15

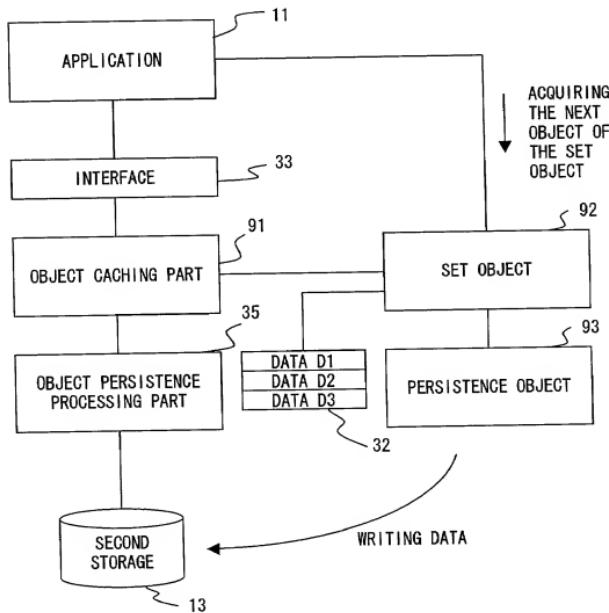
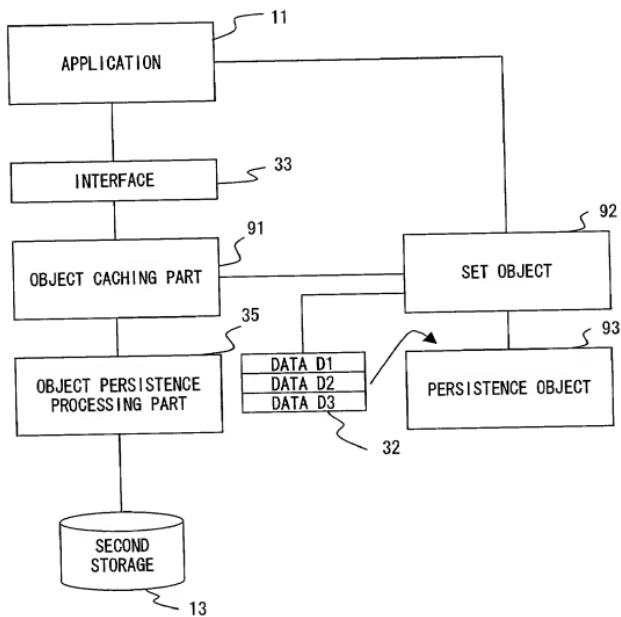


FIG. 16

104260-29524860



F I G. 17

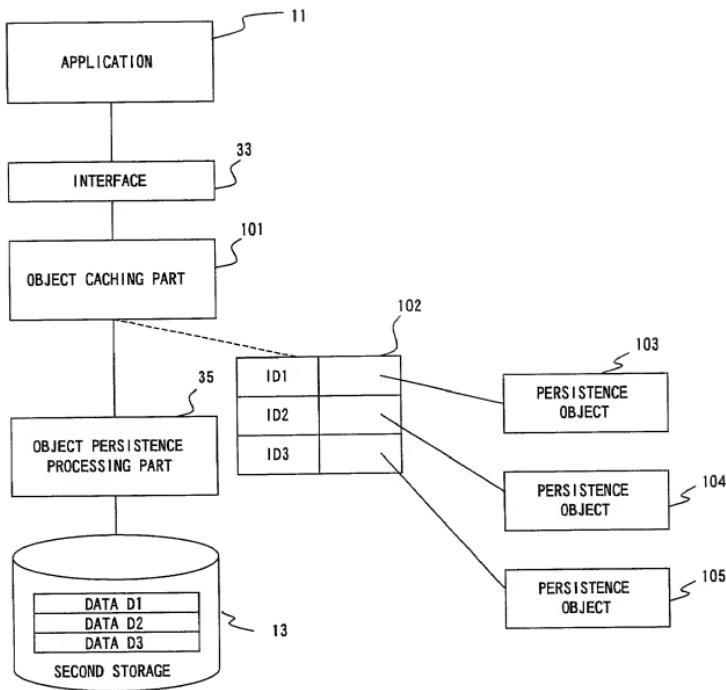


FIG. 18

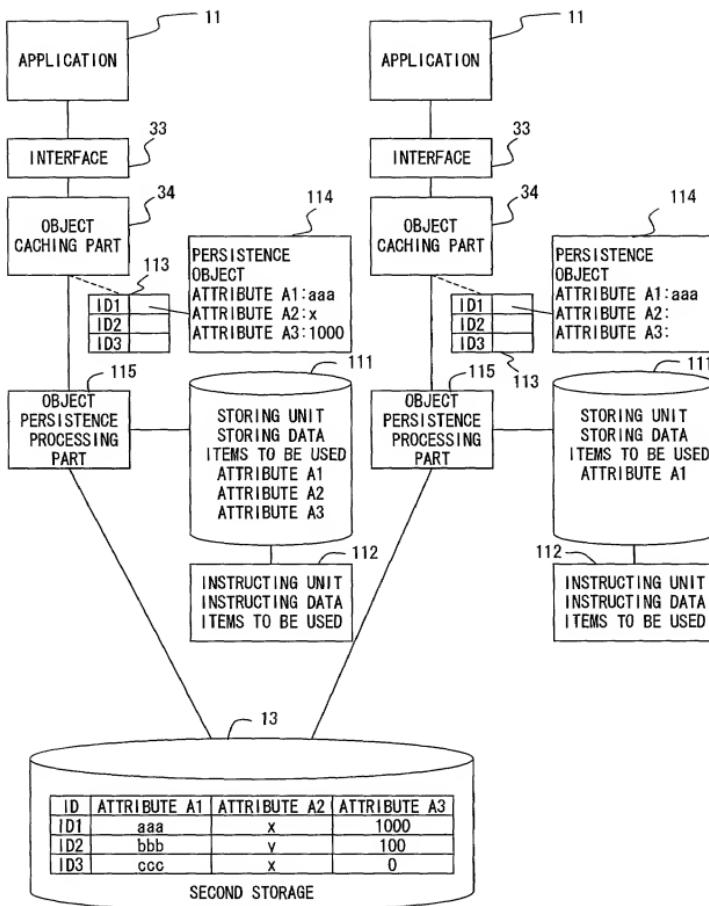


FIG. 19

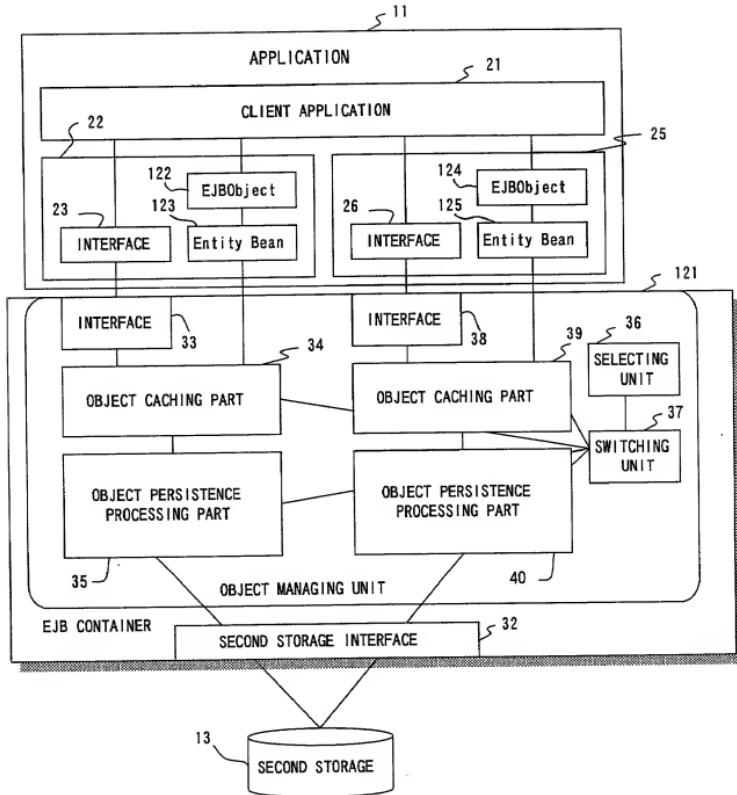


FIG. 20

```
public class $$Bean$$Cache
{
    Hashtable cacheTableHash;
    $$Bean$$Persistence objPersistence;
    $$Bean$$ findByPrimaryKey($$PrimaryKey$$ pk)
    {
        String transactionID = getTransactionID();
        Hashtable cacheTable = (Hashtable)cacheTableHash.get(transactionID);
        $$Bean$$ bean = cacheTable.get(pk);
        if (bean != null) return bean;
        bean = objPersistence.findBean(pk);
        cacheTable.put(pk,bean);
        return bean;
    }
    ...
}
```

```

public class $$Beans$Persistence
{
    DataSource dataSource;

    $$Beans$ findBean($$PrimaryKey$$ pk)
    {
        $$Bean$$ bean = null;
        Connection connection = getConnection();
        String sql = "SELECT $$BeanField$$ FROM $$Table$$ WHERE
                    $$PKField$$ = ?";
        Statement statement = connection.prepareStatement(sql);

        $$foreach $$PKFields$$
        statement.setInt($$Count$$,pk.$$Name$$);
        $$end foreach

        ResultSet rs = statement.executeQuery();
        while(rs.next())
        {
            bean = new $$Bean$$();
            $$foreach $$BeanFields$$
            bean.$$Name$$ = rs.get$$Type$$($$Count$$);
            $$end foreach
        }
        return bean;
    }
    ....
}

```

```
public class OrderBeanCache
{
    Hashtable cacheTableHash;
    OrderBeanPersistence objPersistence;

    OrderBean findByPrimaryKey(OrderBeanPrimaryKey pk)
    {
        String transactionID = getTransactionID();
        Hashtable cacheTable = (Hashtable)cacheTableHash.get(transactionID);
        OrderBean bean = cacheTable.get(pk);

        if (bean != null) return bean;

        bean = objPersistence.findBean(pk);
        cacheTable.put(pk, bean);
        return bean;
    }
    ...
}
```

```
public class OrderBeanPersistence
{
    DataSource dataSource;

    OrderBean findBean(OrderBeanPrimaryKey pk)
    {
        OrderBean bean = null;
        Connection connection = getConnection();
        String sql = "SELECT ID,PRODUCT,QUANTITY FROM ORDERTABLE
WHERE ID = ?";
        Statement statement = connection.prepareStatement(sql);

        statement.setString(1,pk.id);
        statement.setString(2,pk.product);
        statement.setInt(3,pk.quantity);

        ResultSet rs = statement.executeQuery();
        while(rs.next())
        {
            bean = new OrderBean();
            bean.id = rs.getString(1);
            bean.product = rs.getString(2);
            bean.quantity = rs.getInt(3);
        }
        return bean;
    }
    ....
}
```

FIG. 24

```
public class ObjectCacheOption1 extends ObjectCache
{
    Hashtable cacheTableHash;
    ObjectPersistence objPersistence;
    Object findByPrimaryKey(Object pk, BeanDef beanDef)
    {
        String transactionID = getTransactionID();
        Hashtable cacheTable = (Hashtable)cacheTableHash.get(transactionID);
        Object bean = cacheTable.get(pk);
        if(bean != null) return bean;
        bean = objPersistence.findBean(pk,beanDef,"findByPrimaryKey");
        cacheTable.put(pk,bean);
        ....
        return bean;
    }
}
```

FIG. 25

```
public class ObjectPersistenceOption1 extends ObjectPersistence
{
    DataSource dataSource;

    Object findBean(Object pk, BeanDef beanDef, String finderName)
    {
        Object bean = null;
        Connection connection = getConnection();
        String sql = beanDef.getSQL(finderName);
        Statement statement = connection.prepareStatement(sql);

        Enumeration fields = beanDef.getFields();
        while (fields.hasMoreElements())
        {
            FieldDef field = (FieldDef)fields.nextElement();
            setValue(statement,field);
        }

        ResultSet rs = statement.executeQuery();
        while(rs.next())
        {
            bean = beanDef.newInstance();
            fields = beanDef.getFields();
            while (fields.hasMoreElements())
            {
                FieldDef field = (FieldDef)fields.nextElement();
                setValue(bean,rs,field);
            }
        }
        return bean;
    }
    ...
    void setValue(Object bean, ResultSet, FieldDef field)
    {
        Field f = field.getField();
        switch (field.fieldType)
        {
            case FT_INT:
                f.setInt(bean,rs.getInt(field.getColumn()));
                break;
            case FT_LONG:
                f.setLong(bean,rs.getLong(field.getColumn()));
                break;
            ...
        }
    }
}
```

FIG. 26

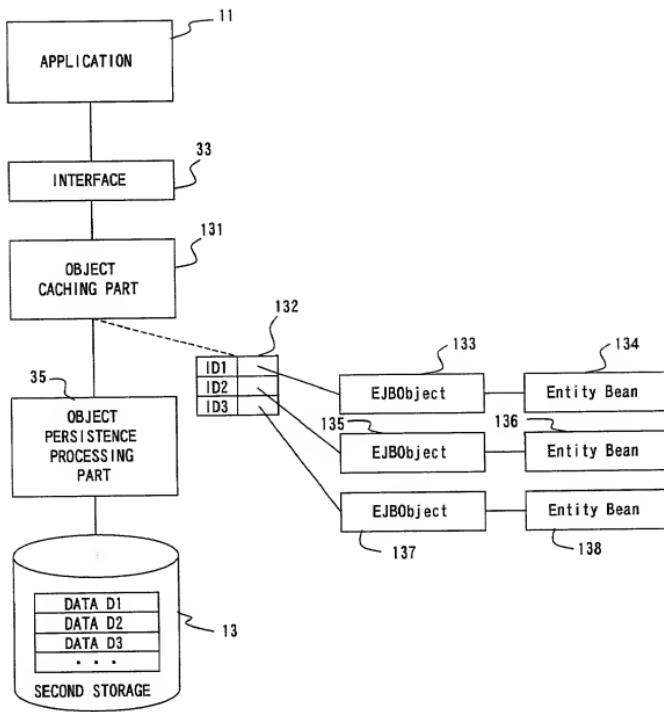
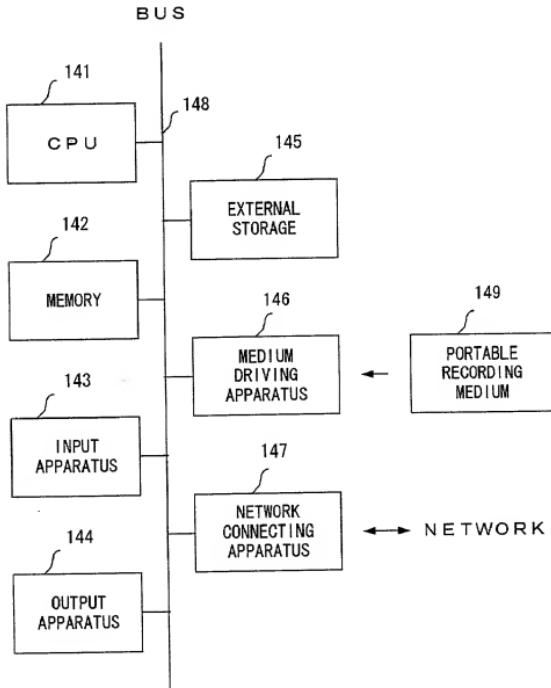
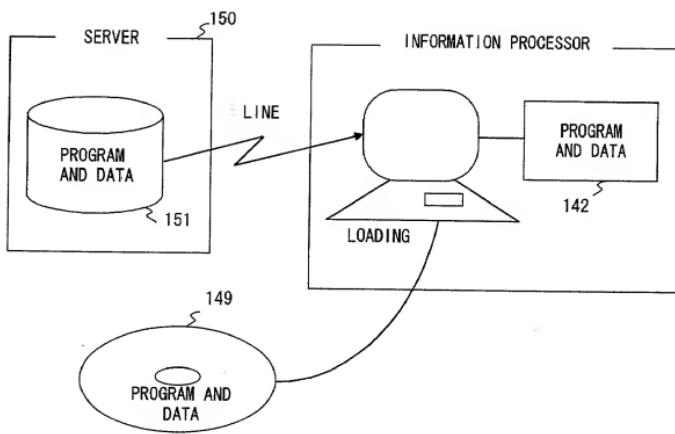


FIG. 27



F I G. 2 8



F I G. 29